

Zend_Acl dynamic loading - Aldemar Bernal

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

Zend Framework: Zend_Acl dynamic loading Component Proposal

Proposed Component Name	Zend_Acl dynamic loading
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Acl dynamic loading
Proposers	Aldemar Bernal
Zend Liaison	Ralph Schindler
Revision	1.0 - 17 November 2007: Proporsal created. 1.1 - 21 January 2008: UCs Created. (wiki revision: 36)

Table of Contents

1. Overview
2. References
3. Component Requirements, Constraints, and Acceptance Criteria
4. Dependencies on Other Framework Components
5. Theory of Operation
6. Milestones / Tasks
7. Class Index
8. Use Cases
9. Class Skeletons

1. Overview

Zend_Acl dynamic loading isn't meant to be a replacement of the current good implementation of Zend_Acl, it is just an update proporsal based on get better performance in the component in medium/large ACL implementations as well as a base to allow the load of persisterced ACLs (from a database or a file or ..).

2. References

- [Zend_Acl online documentation](#)
- [Zend_Acl API](#)

3. Component Requirements, Constraints, and Acceptance Criteria

- This update **will** boost performance in medium/large Acl Implementations.
- This update **will** allow the component to load only the needed Roles/Resources.
- This update **will** allow the component to load data from datasources.

4. Dependencies on Other Framework Components

- Zend_Acl
- Zend_Acl_Role
- Zend_Acl_Resource
- Zend_Acl_Exception

5. Theory of Operation

This update doesn't make any change of current Zend_Acl implementations, it includes only a new method called registerLoader in the Zend_Acl component and a new abstract class called Zend_Acl_Loader_Abstract. The main goal of this new feature is to be able to load dynamically Roles and Resources using an instance of the abstract class. Here an example is given to make clearer the new functionality and its use:

My/AclLoader.php:

```

class My_AclLoader extends Zend_Acl_Loader_Abstract
{
    protected function _init(Zend_Acl $acl)
    {
        /** This method is called after you registered your Loader, so you can load
        default configuration here. */
        $this->getAcl()->addRole(new Zend_Acl_Role('admin'));

        /**
         * This will reroute the method call, if 'configurationstuff' is not found in
        the Acl, the Loader would
         * by default try to call 'setupConfigurationstuffResource'
         */
        $this->setResourceRoute('configurationstuff', '_loadConfigResources');
    }

    protected function _loadGuestRole(Zend_Acl $acl)
    {
        /** This method is called when the guest Role isn't found in the Zend_Acl
        Roles */
        $this->getAcl()->addRole(new Zend_Acl_Role('guest'));
    }

    protected function _loadCartResource(Zend_Acl $acl)
    {
        /** This method is called when the cart Resource isn't found in the Zend_Acl
        Resources */
        $this->getAcl()->add(new Zend_Acl_Resource('cart'));

        /** Setup privileges */
        $this->getAcl()->allow('guest', 'cart', 'index')
            ->allow('guest', 'cart', 'additem')
            ->allow('guest', 'cart', 'removeitem');
    }

    protected function _loadConfigResources(Zend_Acl $acl)
    {
        /** This method is called when the configurationstuff Resource isn't found in
        the Zend_Acl Resources */
        $this->getAcl()->add(new Zend_Acl_Resource('configurationstuff'))
            ->add(new Zend_Acl_Resource('orders'));

        /** This will do recursive call to setupNewsResource */
        $this->getAcl()->allow('admin', 'news', 'add');
    }

    protected function _loadNewsResource(Zend_Acl $acl)
    {
        /** This method was called by loadConfigResources and not from the bootstrap
        file */
        $this->getAcl()->add(new Zend_Acl_Resource('news'));
    }
}

```

At bootstrap:

```

/** Create Acl */
$acl = new Zend_Acl();

/** Register Loader */
$acl->registerLoader(new My_AclLoader());

/**
 * After registered, the method _init of My_AclLoader will
 * be invoked what will load 'admin' role.
 *
 * Ask for any other Zend_Acl Role/Resource would throw an
 * exception since no Role/Resource have been created (only 'admin' Role).
 * But since we registered My_AclLoaded, Zend_Acl will use that
 * class to find a way to load them.
 */

/**
 * After the following call, the roles 'guest' as well as 'cart'
 * resource will be loaded in the Zend_Acl component.
 */
echo $acl->isAllowed('guest', 'cart', 'additem');

/**
 * After the following call, also the resources 'configurationstuff', 'news' and
 * 'orders'
 * will be loaded into the Zend_Acl object.
 */
$acl->allow('admin', 'configurationstuff');

/**
 * Finally, if a role/resource is required but it is not
 * defined in the Zend_Acl object either in the Loader class
 * the default exception will be thrown
 */
$acl->deny('unknown', 'somerresource');

```

So, as you could see in the example above, the Loader class will allow to avoid the problem of loading all ACL at once in bootstrap, and will help to only load the data needed, the results are better response times and less memory usage.

6. Milestones / Tasks

- Milestone 1: [DONE] Create ZF Wiki info.
- Milestone 2: [DONE] Working prototype and some examples. **UCs implemented click [here](#) to download.**
- Milestone 3: Working prototype checked into the incubator supporting use cases.
- Milestone 4: Unit tests exist, work, and are checked into SVN.
- Milestone 5: Initial documentation exists.

7. Class Index

- Zend_Acl
- Zend_Acl_Loader_Abstract

8. Use Cases

I recommend you to download [this](#) file where you can find all UCs implemented for better comprehension of the component use.

9. Class Skeletons

```
]]></ac:plain-text-body></ac:macro>  
]]></ac:plain-text-body></ac:macro>
```